

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-97448

(43) 公開日 平成10年(1998) 4月14日

(51) Int.Cl.⁵

G 0 6 F 12/00

識別記号

5 0 1
5 0 5

FI

G 0 6 F 12/00

5 0 1 A
5 0 5

審査請求 未請求 請求項の数 6 OL (全 7 頁)

(21) 出願番号 特願平9-177283

(22) 出願日 平成9年(1997) 7月2日

(31) 優先権主張番号 6 7 5 0 4 0

(32) 優先日 1996年7月3日

(33) 優先権主張国 米国 (US)

(71) 出願人 597004720

サン・マイクロシステムズ・インコーポレ
ーテッドSun Microsystems, In
c.アメリカ合衆国カリフォルニア州94043-
1100, マウンテン・ビュー, ガーシア・ア
ヴェニュー 2550, エムエス・ピーエイエ
ル1-521

(74) 代理人 弁理士 社本 一夫 (外4名)

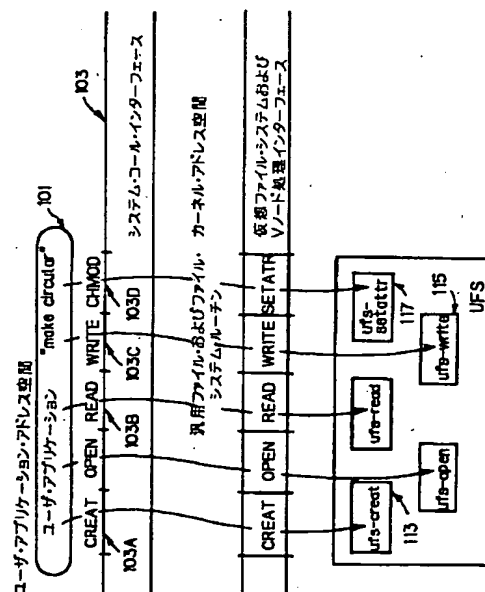
最終頁に続く

(54) 【発明の名称】 コンピュータ・オペレーティング・システムのための循環サイズ制限ファイル方法

(57) 【要約】

【課題】 UNIXオペレーティング・システムまたはその他のレイヤ型またはモジュラー型オペレーティング・システム・アーキテクチャと共に使用する循環ファイルの作成を可能にする装置およびプログラムされた方法を開示する。

【解決手段】 循環ファイルは、iノード・フィールドを用いて、当該ファイルが循環型であることを識別し、ファイルにフィル・データを書き込んで、その作成時に、物理ブロックをファイルに割り当てる。その後、より多くのファイル空間が必要となり、ファイル・サイズが拡大できない場合、ファイルに既に割り当てられているブロックを、再度割り当てる。



【特許請求の範囲】

【請求項1】 ファイル・システムを含み、アプリケーション・プログラムを実行するためのコンピュータ・オペレーティング・システムにおいて、前記ファイル・システムに予測不可能な量の時間評価データを格納する方法であって、
管理用アプリケーション・プログラムから、循環ファイルを作成すべきことを示すアーギュメントを受信するステップと、
インデックス構造を割り当てるステップと、
前記インデックス構造にファイル・サイズを設定するステップと、
前記ファイルに物理ブロックを割り当てるステップと、
ファイルの終端にマークを付けるステップと、
前記インデックス構造に循環マークを設定するステップと、から成ることを特徴とする方法。

【請求項2】 請求項1記載の方法であって、更に、
前記物理ブロックにフィル・データを書き込むステップを含むことを特徴とする方法。

【請求項3】 ファイル・システムを含み、アプリケーション・プログラムを実行するためのオペレーティング・システムを有するコンピュータにおいて、前記ファイル・システムにログ・データを格納する方法であって、
アプリケーション・プログラムから、データを特定のファイルに書き込むべきことを示すアーギュメントを受信するステップと、
前記特定のファイルに対するインデックス構造を発見するステップと、
前記インデックス構造における1フィールドから、前記特定のファイルが循環ファイルであるか否かについて判定を行うステップと、
前記特定のファイルが循環ファイルでない場合、新たに割り当てられた物理データ・ブロックを用いて、前記特定のファイルに前記データを書き込むステップと、
前記特定のファイルが循環ファイルである場合、前記インデックス構造において既に識別されている物理ブロックを再度割り当てるステップと、
前記再割り当てされた物理ブロックにデータを書き込むステップと、から成ることを特徴とする方法。

【請求項4】 コンピュータ・プログラム生産物であって、
コンピュータ・オペレーティング・システムのファイル・システムへのログ・データの格納をコンピュータに行わせる、コンピュータ読み取り可能コードが内部に実装されたコンピュータ使用可能媒体を備え、該コンピュータ読み取り可能コードが、
アプリケーション・プログラムからの、データを特定のファイルに書き込むべきことを示すアーギュメントの受信を、コンピュータに行わせるように構成されたコンピュータ読み取り可能プログラム・コード装置と、

前記特定のファイルに対するインデックス構造の発見を、コンピュータに行わせるように構成されたコンピュータ読み取り可能プログラム・コード装置と、
前記インデックス構造における1フィールドから、前記特定のファイルが循環ファイルであるか否かについての判定を、コンピュータに行わせるように構成されたコンピュータ読み取り可能プログラム・コード装置と、
前記特定のファイルが循環ファイルでない場合、新たに割り当てられた物理データ・ブロックを用いて、前記特定のファイルへの前記データの書き込みを、コンピュータに行わせるように構成されたコンピュータ読み取り可能プログラム・コード装置と、
前記特定のファイルが循環ファイルである場合、前記インデックス構造において既に識別されている物理ブロックの再割り当てを、コンピュータに行わせるように構成されたコンピュータ読み取り可能プログラム・コード装置と、
前記再割り当てされた物理ブロックへのデータ書き込みを、コンピュータに行わせるように構成されたコンピュータ読み取り可能プログラム・コード装置と、から成ることを特徴とするコンピュータ・プログラム生産物。

【請求項5】 コンピュータ・プログラム生産物であって、

コンピュータ・オペレーティング・システムのファイル・システムへの予測不可能な量の時間評価データの格納をコンピュータに行わせる、コンピュータ読み取り可能コードが内部に実装されているコンピュータ使用可能媒体を備え、前記コンピュータ・読み取り可能コードが、
管理用アプリケーション・プログラムからの、循環ファイルを作成すべきことを示すアーギュメントの受信を、コンピュータに行わせるように構成されたコンピュータ読み取り可能プログラム・コード装置と、
インデックス構造の割り当てを、コンピュータに行わせるように構成されたコンピュータ読み取り可能プログラム・コード装置と、
前記インデックス構造に対するファイル・サイズの設定を、コンピュータに行わせるように構成されたコンピュータ読み取り可能プログラム・コード装置と、
前記ファイルに対する物理ブロックの割り当てを、コンピュータに行わせるように構成されたコンピュータ読み取り可能プログラム・コード装置と、
コンピュータに、ファイルの終端にマークを付けさせるように構成されたコンピュータ読み取り可能プログラム・コード装置と、
前記インデックス構造への循環マークの設定を、コンピュータに行わせるように構成されたコンピュータ読み取り可能プログラム・コード装置と、から成ることを特徴とするコンピュータ・プログラム生産物。

【請求項6】 請求項5記載のコンピュータ・プログラム生産物であって、更に、

前記物理ブロックへのフィル・データの書き込みを、コンピュータに行わせるように構成されたコンピュータ読み取り可能プログラム・コード装置を含むことを特徴とするコンピュータ・プログラム生産物。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、一般的にコンピュータ・オペレーティング・システムのファイル・システムの分野に関し、更に特定すれば、かかるファイル・システムにおいて効率的に監査証跡(audit trail)、即ち、「ログ」を格納するための固有のファイル・タイプを、管理ユーザに与えるためのシステムに関するものである。

【0002】

【従来の技術】コンピュータ・システムの多くのアプリケーションは、処理の記録を保持することによって、動作(performance)を追跡し、データの入出力(I/O)の監査を行い、あるいは当該システムの保全性に関する管理または制御を行う必要がある。これらの処理記録は、多くの場合、ログの形態で保持される。その例には、システムおよびアプリケーション・プロセス・メッセージならびにエラー・メッセージを蓄積するメッセージ・ログがある。他の例には、多数のユーザにサービスを提供するタイム・シェアリング・システムにおける、プリント・データ・ログや、メール・ログがある。

【0003】これらのログ・ファイル各々の共通する属性(attribute)は、ログ・ファイルに添加されるデータ量が、管理エンド・ユーザ、あるいは同一のまたは他のコンピュータ上で走るアプリケーションの制御下にあり、したがって、オペレーティング・システムは、どれくらいのデータを格納する必要があるかを正確に判定できないことである。また、これらのログ・ファイルは、多くの場合システムまたはアプリケーションの障害の原因を発見するため、あるいはそれから回復するために使用されるので、ログ内では最新のデータの方が最も古いデータよりも価値が高い。

【0004】

【発明が解決しようとする課題】多くの場合、管理ユーザは、ログ・ファイルが成長すると思われる速度を予測し、偶然性係数(contingency factor)を加算して、全ての予想されるログ・ファイルの合計に対して予め割り当てるべきサイズを決定する。しかしながら、ログ・ファイル毎に固定量の空間を予め割り当てる機構は存在しない。

【0005】システム管理ユーザが蓄積ファイル・サイズの計算を誤って、当該ログ・ファイルには小さすぎるファイル・サイズを指定した場合、ログ内に空間がなくなって、重要な最新のデータ・メッセージ、プリント・データまたはメールが失われる可能性がある。また、管理ユーザが過剰な程に大きなファイル・サイズをログ・

ファイルに指定した場合、古いログ・データを格納することによって物理的空間が使い果たされてしまってディスク記憶装置が使用不能となり、ユーザまたはシステム・アプリケーションに対する新しい物理ブロックの割り当てが不可能となる。

【0006】Samuel J. Leffer et al. の“The design and Implementation of the 4.3BSDUNIX Operating System” (Addison-Wesley Publishing社、1989年発行)の第7章には、ユーザのログ・データを格納する機構に現在利用可能な例が記載されている。

【0007】

【課題を解決するための手段】本発明によれば、先に述べた、指定する空間が小さ過ぎて現情報を失う問題、および指定する空間が大き過ぎて1つのプログラムに全物理ファイル空間を消費させてしまう問題は、新たな循環ファイル・タイプ(circular file type)の作成によって解決される。

【0008】本発明は、特に、論理ファイル空間の全てが満杯の場合でも、最新かつ最も貴重な情報は常に保存されるという点において、ログ用ファイル・システムでは特に有用である。更に、ファイルは循環的であるが、にもかかわらず、ランダムにアクセス可能であり、その中に格納されているデータは、どのような順序でもアクセスすることができ、空間の割り当ておよび再割り当ては全てのログ用アプリケーションに対して自動的に処理することができるので、個別化した調整(preening)および選別(culling)処理はもはや不要となる。更にまた、本発明の循環的なサイズ制限ファイル(circular, size-bounded files)は、何回でもデータの読み出しが可能であり、その空間が最新のデータに必要なまで、読み出すデータは失われない。

【0009】本発明のこれらおよびその他の利点は、ファイル・サイズの定義および物理空間割り当てモジュールに修正を加え、固定量の循環ファイル空間を備えることにより、最新のデータを常に使用可能とし、新たなデータを受信しログに格納する際に、最も古く価値が少ないデータだけを破棄することによって得られる。

【0010】本発明の一特定実施例によれば、ここに開示されるのは、ファイル・システムを含むコンピュータ・オペレーティング・システムと共に使用する方法である。このコンピュータ・オペレーティング・システムは、アプリケーション・プログラムの実行に効果的であり、予測不可能な量の時間評価データ(time valued data)をファイル・システムに格納する方法が開示され、管理用アプリケーション・プログラムから、ファイルを作成すべきことを示すアーギュメントを受信するコンピュータ実装ステップを含む。インデックス構造を割り当て、このインデックス構造にファイル・サイズを設定する。アプリケーション・プログラムはファイル・タイプを変更し、それを循環ファイルに変えることができる。

この循環属性(circular attribute)は、インデックス・ノード構造("inode")に格納される。

【0011】本発明の他の実施例によれば、ファイル・システムを含むオペレーティング・システムを有するコンピュータと共に使用する方法が開示される。このオペレーティング・システムは、アプリケーション・プログラムの実行に効果的であり、ログ・データをファイル・システムに格納する方法が開示され、この方法は、データを特定のファイルに書き込むべきことを示す、アプリケーション・プログラムからのアーギュメントを受信するステップと、この特定のファイルに対するインデックス構造を発見するコンピュータ実装ステップとから成る。この方法は、更に、インデックス構造のあるフィールドから、特定のファイルが循環型か否かを判定するステップと、この特定のファイルが循環ファイルではない場合、新たに割り当てられた物理データ・ブロックを用いて、この特定のファイルにデータを書き込むステップと、特定のファイルが循環ファイルである場合、インデックス構造において既に識別されている物理ブロックを再度割り当てるステップも含む。次いで、ファイル指示子の終端とともに、データを物理ブロックに書き込む。

【0012】本発明の上述のおよびその他の特徴ならびに目的、ならびにそれらを達成する方法は、添付図面に関連付けて以下の好適実施例の記載を参照することによって、より明白となり、本発明自体も最良に理解されよう。

【0013】

【発明の実施の形態】本発明をUNIXの例に実施可能な構造およびモジュールについて、これより図1を参照しながら説明する。図1は、典型的なUNIXファイル・システムの一部である、様々な従来技術のシステムを纏めたものである。アプリケーション・プログラム101はシステム・コール・インターフェース103を用いて、ファイル・システム105に実装されているファイル・システム関数をコールし、データの格納および読み出し機能を実行する。これについては、Samuel J. Leffler et al.による"The Design and Implementation of the 4.3BSD UNIX Operating System"と題する文献の第6章に、より詳細に記載されており、図1の殆どはこの文献から引用したものである。この本は1989年にAddison-Wesley Publishing社によって発行され、その内容は、この言及により、本願にも含まれるものとする。ファイル・システム・ブロック105について、本発明のモジュールおよび構造をより詳しく示す図2を参照しながら説明する。

【0014】図2の最上部には、ログ・データを生成するユーザ・アプリケーション101が示されており、多数の標準ファイル処理システム・コールを呼び出している。これは、CREAT103Aを呼び出し、通常のUNIXの方法でファイルを作成する。このコールは適切なファイ

ル・システム・ルーチン、即ち、ufs-creat 113に渡される。これによって、Leffler et al.のような前述の参考文献に記載されているファイルの作成が可能となる。次に、ufs-write 115に指令(dispatch)される書き込みコール103Cを繰り返し発行することによって、このファイルを所望のサイズまで成長させる。最後に、ユーザ・アプリケーションはCHMOD103Dを呼び出し、このファイルに新しい循環属性(circular application)を設定する。これは、ufs-setattr 117ルーチンに渡され、ファイル毎のデータ構造およびiノード(インデックス・ノード構造)を更新し、今後の読み出し103Bおよび書き込み103C処理のための新たな循環的割り当ておよび再割り当て方針(policy)を形成する。この循環ファイルは、この時点において、循環ファイルとして使用する準備ができています。

【0015】図3は、循環割り当てモジュール109の部分を示す。これは、現ログ・データを格納するために他のブロックが必要となった場合に、EOFを含むブロックに続く物理ブロックを再度割り当てるように動作する。部分133は、汎用ファイル・システム・ルーチンを用いて、iノード・テーブルにおいてまたはディスクから読み込むことによって、iノードを見つけ出す。部分135は、ファイル・タイプ・ビットを検査し、当該ファイルが循環ファイルであるか否かについて判定する。循環ファイルではない場合、汎用ファイル・システム・ルーチンは、部分136においてその書き込みプロセスを完了する。当該ファイルが循環型である場合、部分137において現物理ブロックにアクセスし、EOFを見つけなければならない。

【0016】現物理ブロックが未だ空間を有する場合、部分139において、ログ・データをブロック・バッファに移動させ、部分145においてディスクに書き込む。ブロックが満杯になった後に、未だ格納すべき現ログ・データが残る場合、ブロック・マッピング・ルーチンによって別のブロックを割り当てなければならないが、新たなブロックを割り当てるのではない。代わりに、部分141では、Lefflerの参考文献の第207および208ページに記載されている汎用割り当てBMAPルーチンを迂回して、論理ブロック・ポインタを次の連続位置に移動させ、ブロック・テーブルにおいて識別されている物理ブロックを、新たな論理ブロックに再度割り当て、検索し、そして部分143において消去する。消去した後、部分145において残りのログ・ファイルをバッファに移動し、部分147において論理EOFマーカを更新し、ディスクまたはその他の持続的コンピュータ大容量記憶媒体に書き込む。

【0017】本発明がその有用性を見出す環境は、汎用分散型計算機システムが含まれ、かかるシステムにおいては、汎用コンピュータ、ワークステーションまたはパーソナル・コンピュータが、種々のタイプの通信リンク

によって、クライアントサーバ構成で接続されており、多くの場合オブジェクトの形態のプログラムおよびデータが、システムの他のメンバによるアクセスおよび実行のために、システムの様々なメンバに使用可能となっている。図4は、汎用ワークステーション・コンピュータの一例を示し、プロセッサ201は、I/O部202、中央演算装置即ちCPU203およびメモリ部204を有するものとして示されている。I/O部202は、キーボード205、視覚表示装置206、ディスク記憶装置209およびコンパクト・ディスク・リード・オンリ・メモリ（CDROM）駆動装置207に接続されている。CDROM装置207は、典型的に、図1ないし図3に関して既に述べたプログラム生産物210を収容する、CDROM媒体208を読み込むことができる。これらのコンピュータ・プログラム生産物は、本発明の装置および方法を実行する機構を内蔵し、かかるシステムのメモリ204、またはディスク記憶装置209上、またはCDROM208上に置くことができる。

【0018】以上、本発明をUNIX型アーキテクチャにおけるその好適実施例に関して説明したが、上述の実施態様の構造およびプログラムにおいて、特許請求の範囲によって規定される本発明の精神および範囲から逸脱することなく、他のレイヤ型またはモジュール型オペレーティング・システム・アーキテクチャにおける本発明の応用を含む、種々の変更が可能であることは、コンピュータ・システム設計における当業者には認められよう。

【図面の簡単な説明】

【図1】本発明による、オペレーティング・システムおよび媒体装置の様々なレベルに対するアプリケーション・プログラムの関係を示す代表的ブロック図。

【図2】本発明の代表的実施例による、図1のファイル・システムの構造を示す簡略表現図。

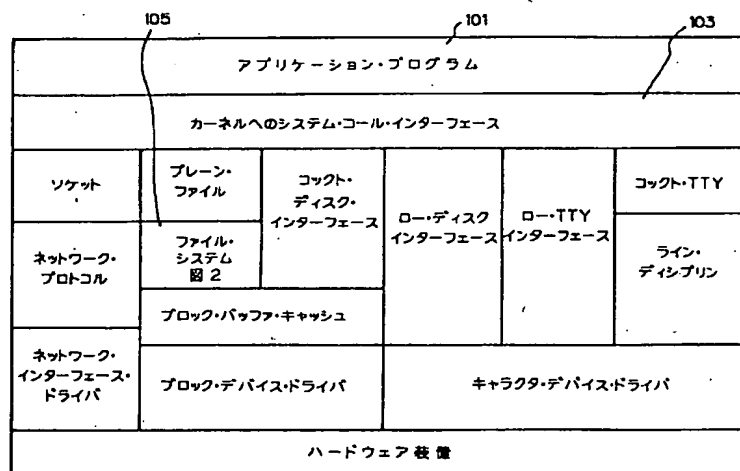
【図3】本発明による方法を表わす論理フローチャート。

【図4】本発明の動作環境の一部を形成する汎用コンピュータの簡略図。

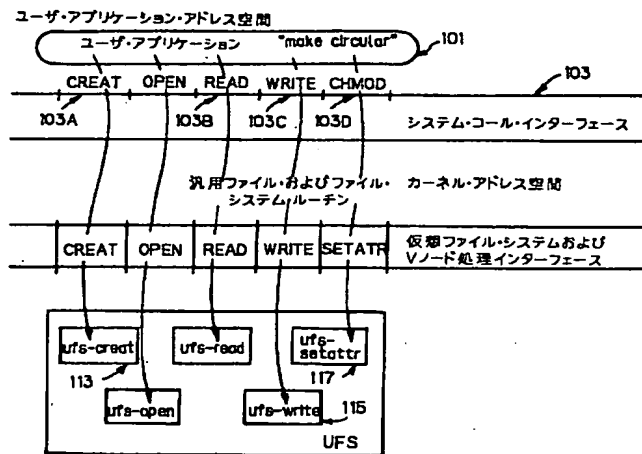
【符号の説明】

- 101 アプリケーション・プログラム
- 103 システム・コール・インターフェース
- 105 ファイル・システム
- 109 循環割り当てモジュール
- 201 プロセッサ
- 202 I/O部
- 203 中央演算装置
- 204 メモリ
- 205 キーボード
- 206 視覚表示装置
- 207 CDROM装置
- 208 CDROM媒体
- 209 ディスク記憶装置
- 210 プログラム生産物

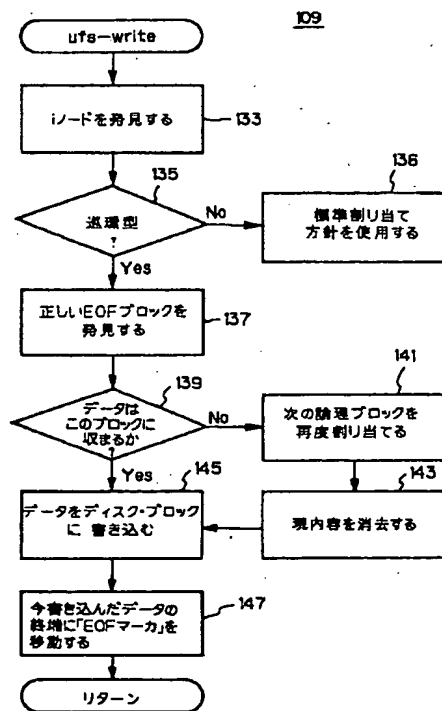
【図1】



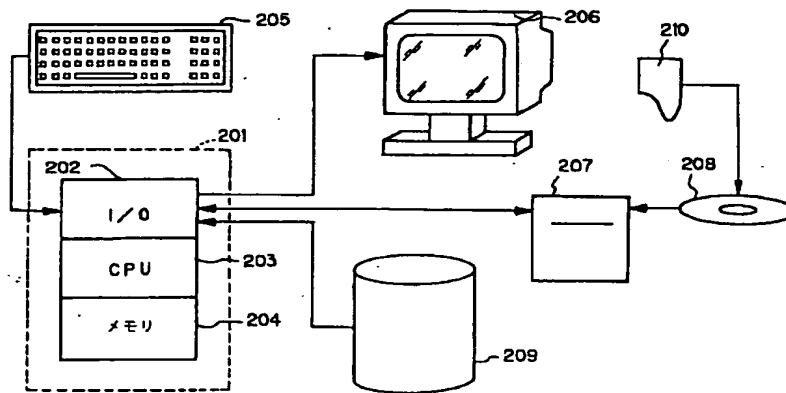
【図2】



【図3】



【図4】



フロントページの続き

(71)出願人 597004720
2550 Garcia Avenue, MS
PAL1-521, Mountain V
iew, California 94043-
1100, United States of
America

(72)発明者 スティーブン・ティー・セネター
アメリカ合衆国コロラド州80920, コロラ
ド・スプリングス, ウェストミンスター・
ドライブ 8625